**Team Introduction:**

This project is done by Zijian Huang and Lubing Wang, two PhD students from Statistics department.

**Project Objective:**

In this project, we focus on building a financial trading portfolio. Financial trading is the buying and selling of financial assets, also called financial securities. People trade a variety of financial instruments, including equities: shares of stocks representing ownership of companies, bonds: debt instruments issued by the government or corporations, forex or foreign exchange market of currencies, commodities such as gold, silver, and oil, and cryptocurrencies like Bitcoin. However, in our case, our target assets are the stock in the US market.

People trade to make a profit by taking calculated risks. A trader makes a profit when buying a security at a lower price and selling later at a higher price, known as going long. Conversely, they may sell a (borrowed) security at a higher price and buy it back at a lower price, known as going short. As a retail trader, we focus on the long side.

Trading typically has a shorter holding period, ranging from days to months. Investing has a longer time horizon, ranging up to years or even decades. Trading focuses on short-term market trends and tries to profit from volatility and price fluctuations.

Portfolios are a bundle of individual stocks with different weights for each position. The return of a portfolio is a linear combination of the weights and returns of each position. We are using discrete returns instead of log returns.

To select target stocks, sentimental analysis is applied on several most popular financial communities in reddit. Then we use Efficient Frontier Optimization proposed by Harry Markowitz to optimize our portfolio using stocks picked by sentimental analysis.

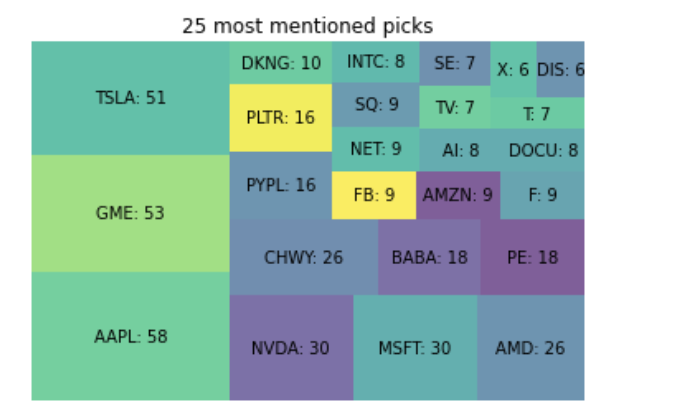
**Method – Sentimental Analysis**

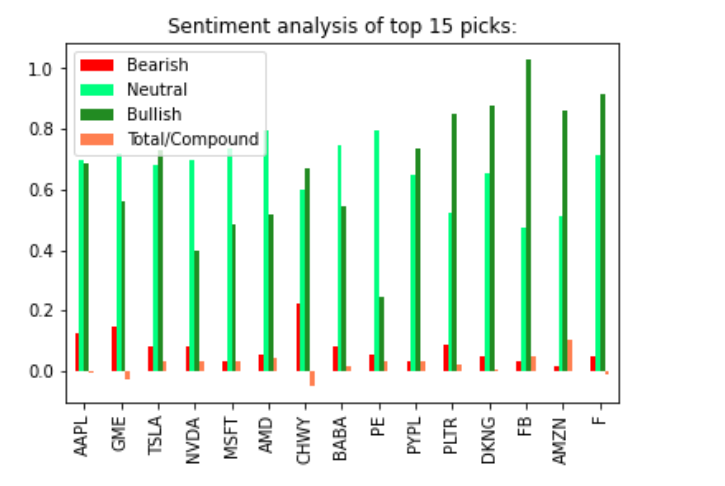
We use Vader SentimentIntensityAnalyzer to conduct the sentimental analysis with several most popular finance communities in reddit so that we can build the stock selector.

We generated the data.py file and import it as a module. We define all the stock ticker that we want to research. The other important thing is new words. It is a dictionary where the keywords that are used to qualify a sentiment towards a stock and the value tell us how bullish it is.

There are some pre-defined parameters. First, we define which subreddit we want to research, in our case, we involved wallstreetbets, stocks and investing, three large finance communities on Reddit. Then we are going to filter certain posts that have this flare, such as Daily Discussion, Weekend Discussion and Discussion. We decided to pick top 500 hottest most recent posts within around 1 week from those communities. In order to guarantee the quality of the posts, we filter the ones with only upvote ratio greater than 70%. For the comments within the posts, they also need to have a minimum number of upvotes which is 2. It was defined that 25 most mentioned tickers should be generated in order to perform the sentiment analysis and we chose top 15 to do the sentiment analysis. Overall, we have 11289 comments in 179 posts.

The result is as follows:



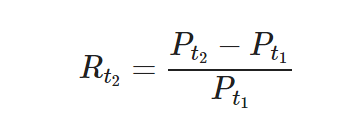


Based on the first table, it showed 25 most mentioned tickers. We try to build a portfolio based on those stocks. Since there are too many stocks and some of them may turn out to short, the score could help us further filter the stocks. After considering the compound score, we discard the stocks with negative scores. Moreover, PE is a kind of special assets, thus, it should be discarded from our final portfolio as well. Overall, our portfolio included TSLA, NVDA, MSFT, AMD, BABA, PYPL, PLTR, DKNG, FB and AMZN. Those 10 stocks seem to be ideal invest target for the short term.

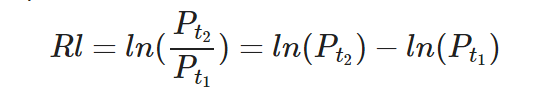
**Method – Portfolio Optimization**

Financial returns are generally derived from stock prices, and are expressed as percentages in decimal form. There are two common types of financial returns: discrete returns (also called simple returns), and log returns (also called continuous returns). In our project, we used mostly discrete returns at the daily frequency. Log returns are used for some advanced formulas and financial models in finance.

There is a very simple formula to calculate simple returns, which is simply today's price minus yesterday's price , divided by yesterday's price. This gives you a percentage gain or loss for the day.

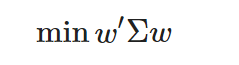


For log returns, what is important to understand though is that log returns aggregate across time, while discrete returns aggregate across assets. Since you are building portfolios with multiple assets, discrete returns make the most sense to our project.

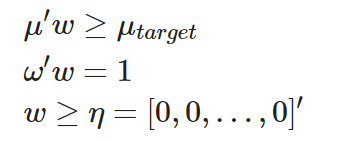


Harry Markowitz came up with Efficient Frontier Optimization in 1952, which provides a solid framework for combining stocks in a portfolio. The key insight is that by combining assets with different expected returns and volatilities, one can decide on a mathematically optimal allocation.

Mean-variance optimization problem

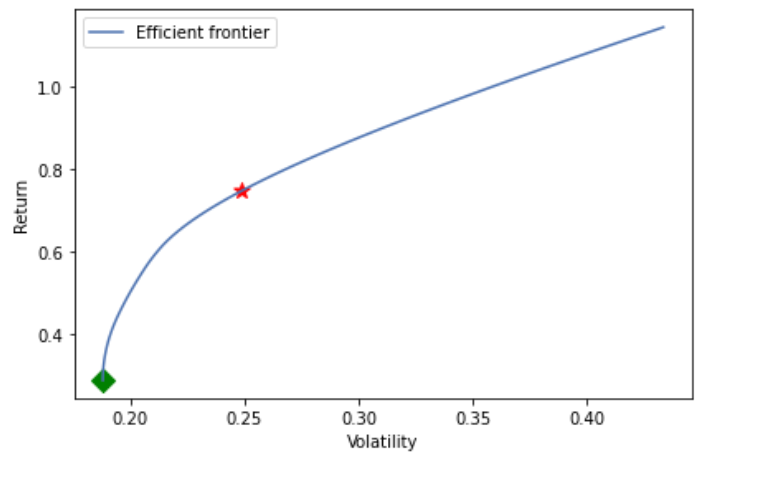


with constraints:



The Efficient Frontier Optimization is a constrained optimization problem, in which you try to minimize portfolio variance by setting the weights. The top line is in fact the portfolio variance. Its weights transposed, times the covariance matrix sigma, times the weights again. The "subject to" conditions simply state that the expected return, or weights times mu, should be at least some target level of return. Then, it says that the weights should sum up to 1, and that at least some of the weights should be positive.

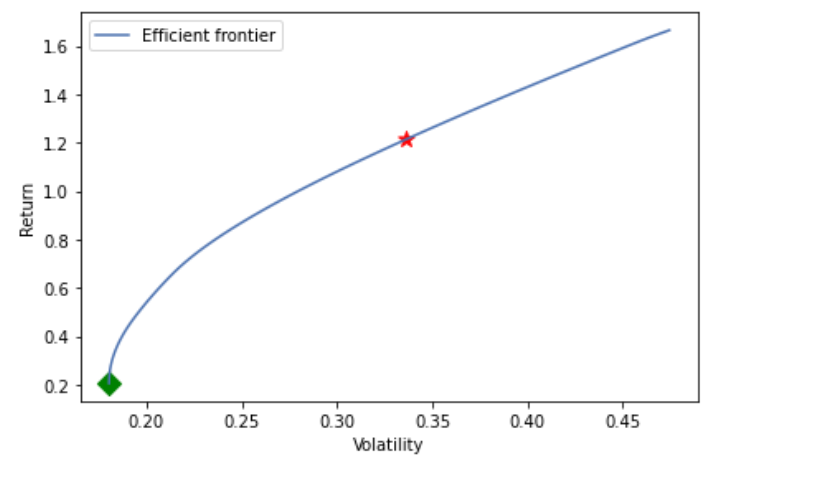
When we change the target return in the optimization problem, the weights w that form the optimal solution, change. Thus, the solution to the Markowitz' optimization problem, is a collection of portfolios, which we call the efficient frontier. The line in the graph represents that frontier. The efficient frontier is thus the set of optimal portfolios that offers lowest risk for a given level of expected return and vice versa.



μ and σ are normally calculated by the returns. The red star represents the point on the efficient frontier with the max sharpe ratio. On the other hand, the green diamond represents the point on the efficient frontier with min volatility. Moreover, we could check the performance for each case.

Efficient frontier optimization requires knowledge of the expected risk Sigma and expected returns mu. In practice, these are rather difficult to know with any certainty. The best we can do is to come up with estimates, for example by extrapolating historical data. But that is where we go wrong. If history would repeat itself perfectly, we would all be able to predict financial markets and stock movements. The truth is, the mean historic returns, or the historic portfolio variance are not perfect inputs and do not reflect future expected risk and return perfectly. Hence the resulting weights of our optimization problem, would have worked well in the past, but we have no guarantee that it will work well in the future.

We need to think about better measures for expected risk and return. A possible improvement is to use exponentially weighted risk and return. It assigns more importance to the most recent data, and thus aims to improve the estimates.

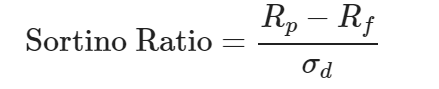


We can do the same for volatility. The exponential covariance matrix also gives more weight to recent data when calculating covariance, in the same way that the exponential moving average is calculated.

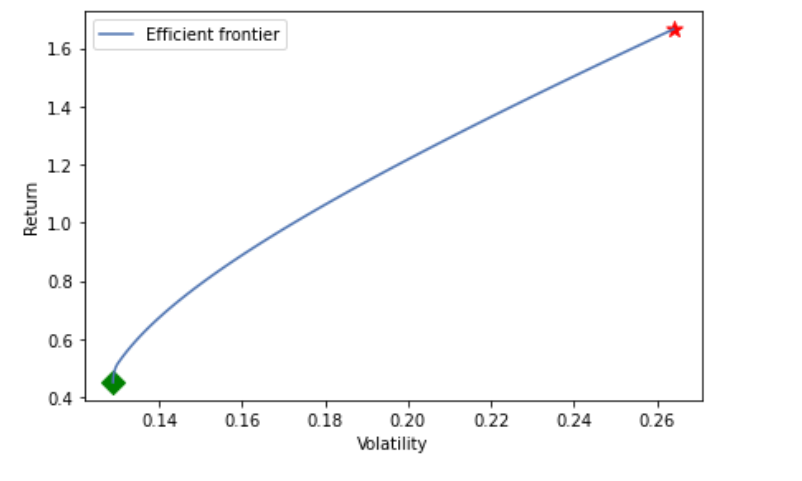
The efficient frontier is calculated by exponential weighted μ and σ. The red star represents the point on the efficient frontier with the max sharpe ratio. On the other hand, the green diamond represents the point on the efficient frontier with min volatility. Moreover, we could also check the performance for each case.

When we use measures such as standard deviation and variance for volatility, it measures all deviations from the mean, so both upside and downside deviations. However, investors typically more focus on the downside in their portfolios. They do however worry about negative returns. That suggests that a good risk measure should perhaps focus on the potential losses, rather than overall fluctuations measured by volatility.

That's where the Sortino ratio comes in. The Sortino ratio is a variation of the Sharpe ratio. It differentiates harmful volatility from overall volatility by using the asset's standard deviation of negative portfolio returns only. So instead of the normal standard deviation, in the Sortino ratio you calculate the standard deviation of the negative returns only, called downside risk.



where the denominator is the standard deviation of the downside. For that ratio you calculated the variance of the negative returns only, as a way to measure downside risk. Here we use the semi-covariance matrix for the portfolio optimization problem.



So, the max sharpe ratio will lead to max sortino ratio instead. This time the max sortino ratio method just chooses the NVDA as our only choice. It seems really aggressive trading strategy.

**Result – Backtesting**:

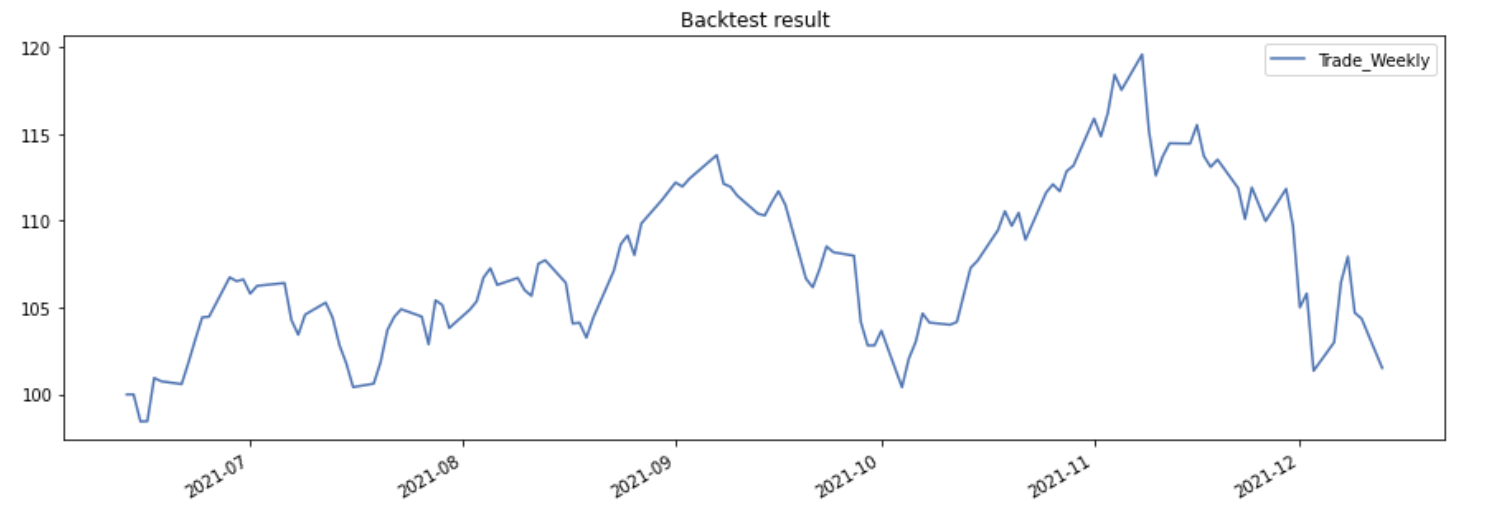
Backtesting is a way to assess the effectiveness of a strategy by testing it on historical data. The test result is evaluated to determine how it would have performed if used in the past, and whether it will be viable for further trading.

There are four steps to define and backtest a strategy. First, we obtain historical price data of the assets we are going to trade. Second, we define the strategy. Next, we backtest the strategy with the historical data, and finally we evaluate the result.

First, we download the "Adjusted Close" prices from Yahoo Finance by all chosen stocks from '2021-6-14' to '2021-12-14'. The "Adjusted Close" price is adjusted for events like corporate actions such as stock splits, dividends, etc. Moreover, it is also the price we use before.

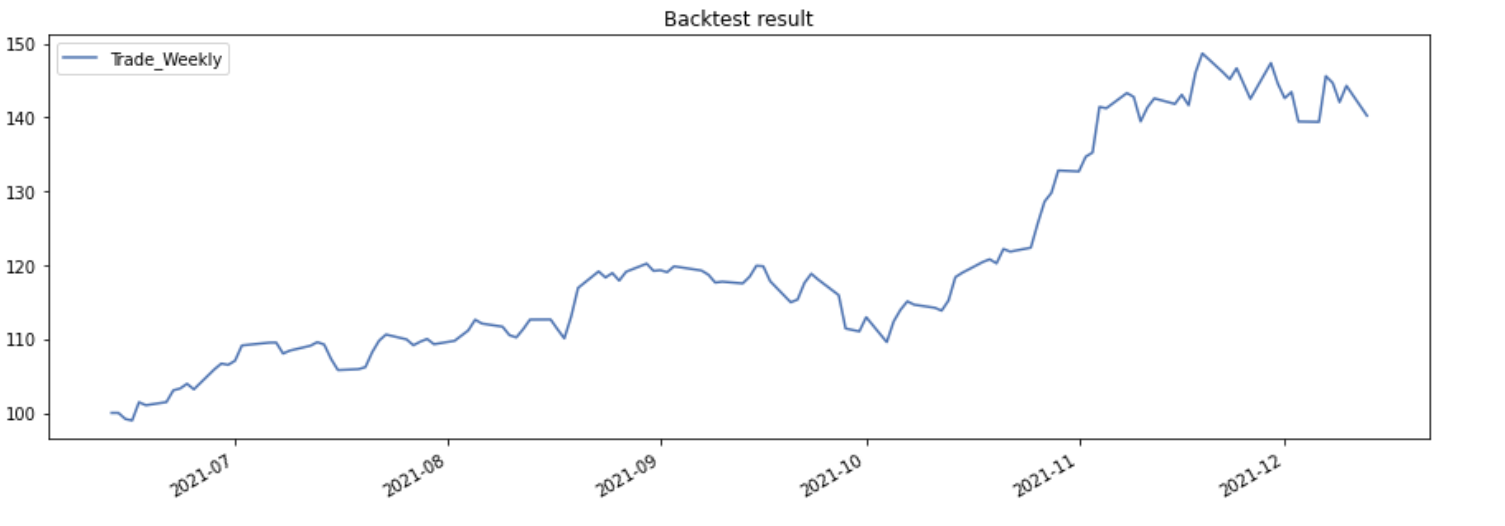
The "Strategy" contains trading logics by combining various "algos". It allows us to easily create strategies by mixing and matching different algos, each of which acts like a small task force that performs a specific operation. Within "Strategy" we first assign a name. Then we define a list of algos in the square brackets. The first "algo" specifies when to execute trades. Here we specify a simple rule to execute trades every week using "RunWeekly". The second "algo" specifies what data the strategy will apply to, for simplicity we apply to all the data using "SelectAll". The third "algo" specifies, in the case of multiple assets, what weights apply to each asset. Here we try different kinds for weighted in order to test which portfolio performs the best. The last "algo" specifies that it will re-balance the asset weights according to what we have specified in the previous step. We now have a strategy that will execute trades weekly on a portfolio that holds several stocks.

Benchmark model: Here we use equal weighted method as a benchmark in order to compare the performance with other weighted we gained before. It seems not a good combination. Since the return, sharpe ratio and sortino ratio are all pretty small.



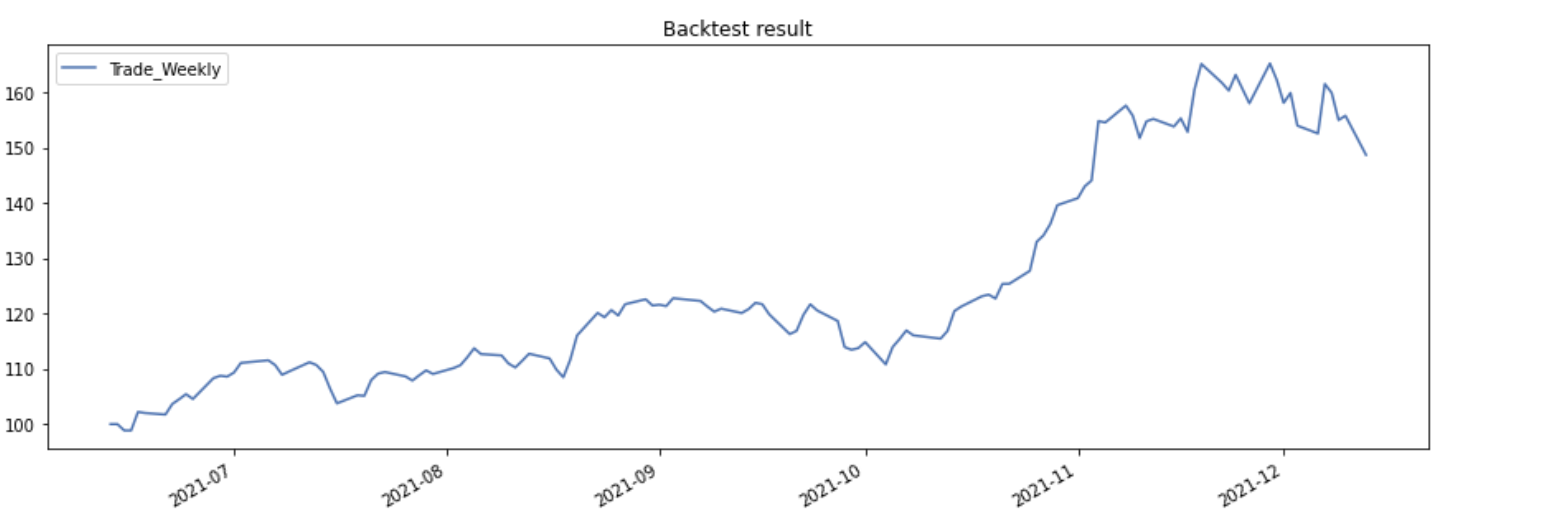
Normal mean and variance:

This backtest is based on the normal calculated μ and σ. We choose the weight of msft=0.66455 and nvda=0.33545. It turns out to offer us really good results. Total return 40.22%, sharpe ratio is 2.96 and sortino ratio is 5.72. The volatility is lower than the benchmark.



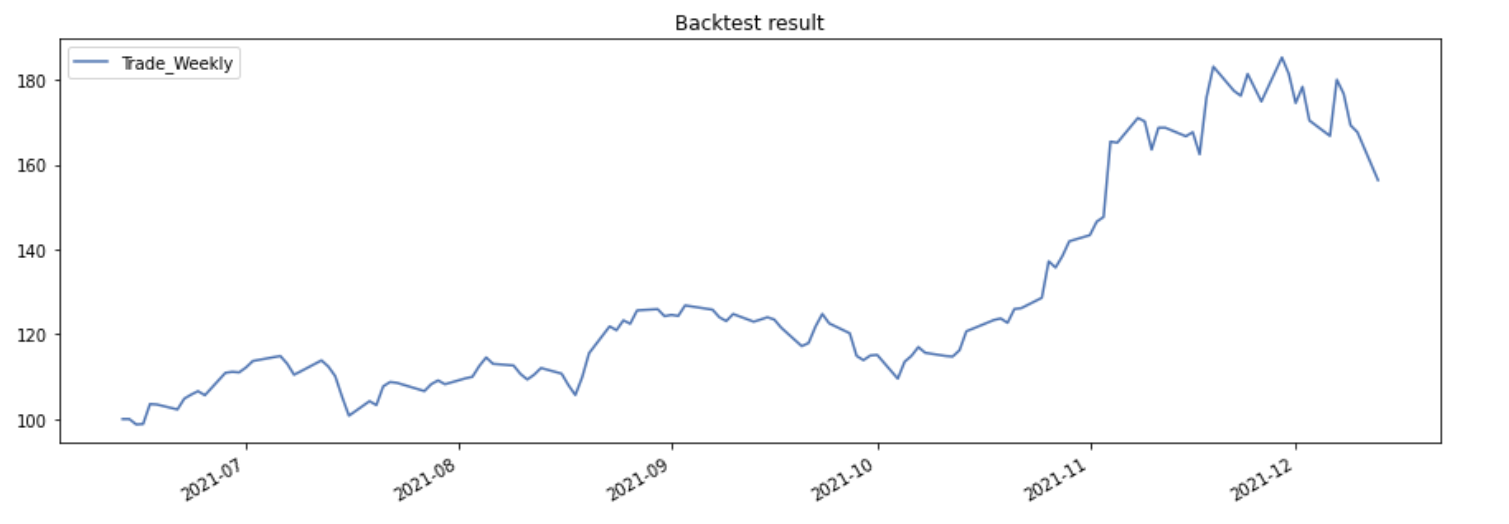
Exponential mean and variance

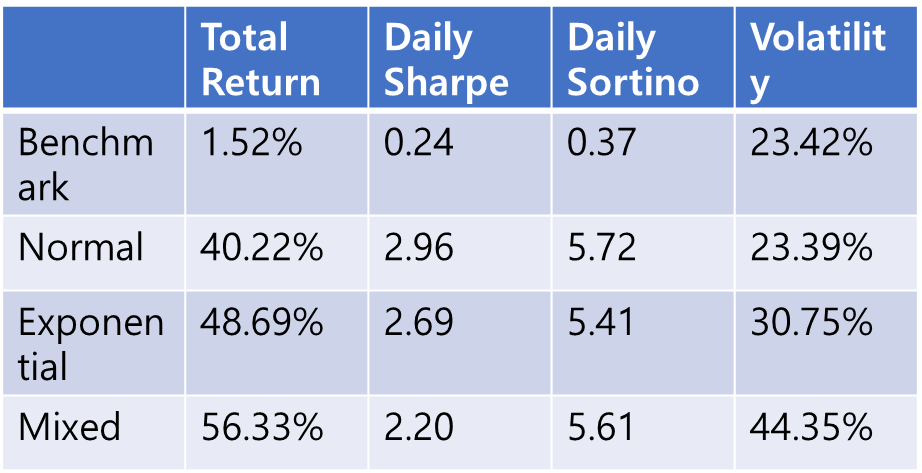
In this case, we use exponential weighted μ and σ to conduct the backtest. We choose the weight of msft=0.34813, nvda=0.58744 and tsla=0.06443. It also turns out to give us good results. This time the total return is higher than the previous one and it is near 50%. However, the sharpe ratio and sortino ratio are slightly lower than the previous one.



Exponential mean and semi-covariance:

In this case, we use exponential weighted μ and semi-covariance to conduct the backtest. It only has nvda. Although the total return is the highest among all other combination. The volatility is also highest among the others and it is even worse than the benchmark. Since it just picks nvda as our only target asset.





**Conclusion**

According to modern portfolio theory, pioneered by Harry Markowitz in 1952, there is an efficient frontier of portfolios, each with the highest expected return for a given level of risk. We select the tangency portfolio, which is the portfolio with the highest Sharpe ratio. The tangency portfolio is known as the max sharpe ratio, or MSR portfolio.

We pick the stock by using sentiment analysis for the largest finance communities in Reddit. Then, we set the equal weighted stocks as our benchmark and the performance is relatively poor. However, MSR portfolio turn out to have obviously better results by different μ and σ. MSR portfolio by using exponential weighted μ and σ should become a ideal portfolio. Since it has higher total return and lower volatility compared with other two portfolios.

Since our report is based on the short-term information, so it is crucial to rerun the code every week in order to detect the new hottest stock in the market and make the new portfolios.

The code refers to the github page: <https://github.com/yellowzijian/Math5670_Final_Project/blob/main/Final_Project.ipynb>

Video Link:

Channel 1:

<https://www.youtube.com/watch?v=kdaRcCadOPE>

Channel 2:

https://www.youtube.com/watch?v=bwwjoN1DJjA